



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/001,478	11/01/2001	Craig Nemecek	CYPR-CD01213M	6435

7590 01/30/2008
WAGNER, MURABITO & HAO LLP
Third Floor
Two North Market Street
San Jose, CA 95113

EXAMINER

PROCTOR, JASON SCOTT

ART UNIT	PAPER NUMBER
----------	--------------

2123

MAIL DATE	DELIVERY MODE
-----------	---------------

01/30/2008

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)
	10/001,478	NEMECEK ET AL.
	Examiner	Art Unit
	Jason Proctor	2123

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 09 November 2007.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1,2,4-10,12 and 14-23 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1,2,4-10,12 and 14-23 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date <u>11/9/07</u> | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

Claims 1-2, 4-10, 12, and 14-23 were rejected in the Office Action entered on 17 September 2007.

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 31 October 2007 has been entered.

Applicants' response submitted on 31 October 2007 has amended claims 1, 2, 4, 5, 7-10, 12, 15, 16, 18, 19, and 20. Claims 1, 2, 4-10, 12, and 14-23 are pending in this application.

Claims 1, 2, 4-10, 12, and 14-23 are rejected.

Priority

1. This Application contains a claim for the benefit of priority to U.S. Provisional Application No. 60/243,708 filed 26 October 2000. The provisional application has been reviewed and priority is denied, because the provisional application does not appear to enable the claimed invention as required under 35 U.S.C. Section 112, first paragraph. See 35 U.S.C. § 119(e)(1).

For example, the provisional application contains a set of 'powerpoint-style' drawings and datasheets describing desired features for a microcontroller or a 'system-on-chip,' but this material does not appear to contain either the text description or the drawings found in the

Application. In particular, no part of the provisional application appears to disclose the method steps shown in the Application at Fig. 7.

Information Disclosure Statement

2. The information disclosure statement filed 9 November 2007 fails to comply with 37 CFR 1.98(a)(2), which requires a legible copy of each cited foreign patent document; each non-patent literature publication or that portion which caused it to be listed; and all other information or that portion which caused it to be listed. It has been placed in the application file, but the information referred to therein has not been considered.

The Information Disclosure Statement claims to filed "before the mailing of the first Office action on the merits" which is incorrect. This Information Disclosure Statement was filed "before the mailing of a first Office action after the filing of a request for continued examination under § 1.114."

The Information Disclosure Statement claims that "copies of publications listed on Form PTO-1449 from prior application Serial No. 09/975,105 filed on October 10, 2001, of which this application claims priority under 35 U.S.C. § 120, are not being submitted pursuant to 37 CFR § 1.98(d). The Examiner has considered every such publication that can be located in the parent '105 application. However, publications listed as Y2-Z2, A3-R3, W3-Z3, and A4 are not found in this application or the '105 application. Accordingly, these references have not been considered by the Examiner.

Further, references S3-V3 identify US Patent applications which are not publications suitable for printing on the front page of a patent, should a patent issue from the instant application. The Examiner has considered and initialed these references, but has lined through them so that they will not be printed on any patent.

Response to Arguments – 35 USC § 103

3. In response to the previous rejections under 35 U.S.C. § 103, Applicants argue primarily that:

Dictionary meanings of “dummy code” clearly show that the ordinary and customary meaning of “dummy code” is different from “booting code” recited in Claim 1. Unlike “booting code” which is used to boot the microcontroller, “dummy code” does not perform any functions to boot the virtual microcontroller. “Dummy code” is used in one embodiment to synchronize the virtual controller with the microcontroller using a timing loop.

The Examiner responds as follows.

The Examiner thanks Applicants for providing clear factual support for their interpretation of the claim language. The Examiner has fully considered this argument and finds it persuasive. Accordingly, the previous rejections under 35 U.S.C. § 103 are withdrawn. New grounds of rejection are entered below.

4. Applicants also argue that:

Coker explicitly states that “[t]he shadow system receives its input data from the input registers of the target-ECS and stores the input data in its RAM,” See Column 2, lines 61-63 (emphasis added) and that “[t]he target-ECS is connected by a one-directional bus with an interface means comprising means for receiving input signals from the target—CS and converting the input signals to generic input signals, means for converting the generic input signals to unique input events and providing said unique input events at the first opportunity to a connected shadow system or host system.” See Column 4, lines 9-16. The passages in Coker at least indicate that the target-ECS may be accessible by the shadow system. Unlike the limitation of Claim 1, Coker does not teach that the target-ECS is inaccessible by the shadow system. Instead, Coker teaches that the target-ECS can be accessed by the shadow system throughout their operations.

The Examiner respectfully traverses this argument as follows.

Claim 1 recites that “the set of boot code is inaccessible to the virtual microcontroller” (emphasis added). Coker does not teach, and Applicants do not argue, that Coker's system transmits, receives, or in any way accesses any form of boot code. Applicants' argument is directed to Coker's teachings regarding input signals, which are very different from boot code. The Examiner maintains the position that Coker contains no disclosure whatsoever to suggest that the shadow system can, in any way, "access" the boot code in the microcontroller.

This argument has been fully considered but has been found unpersuasive.

5. Applicants also argue that:

In addition with respect to currently amended Claim 2, Applicants respectfully assert that Rosenberg or Coker or their combination fails to teach or fairly suggest the limitation of “after completion of the simultaneous halting” as recited by Claim 2.

The Examiner respectfully traverses this argument as follows.

It is unclear what basis exists for Applicants' argument.

Claim 1 recites “simultaneously halting both the microcontroller and virtual microcontroller” which has previously been shown in Rosenberg [*A significant issue for debuggers on multiprocessor systems is how or whether other processors stop when a fault occurs in one. On SIMD architectures, by definition, all processors operate in lock step and so this is guaranteed.*” (page 45, first full paragraph)]. Applicants have not argued that Rosenberg fails to teach this limitation. Applicants have not argued that the combination of references for the purposes of § 103 is improper.

Applicants' amendment duplicated this limitation in claim 2. However, according to 37 CFR 1.75(c), a dependent claim includes the limitations of the parent claim. Duplication of claim language is not necessary.

In conclusion, Rosenberg teaches simultaneously halting the microcontroller and virtual microcontroller (page 45, first full paragraph). Coker teaches copying register contents from the microcontroller to the corresponding registers in the virtual microcontroller (column 2, lines 61-63). Applicants have not traversed either of these teachings nor the combination of references.

Applicants' argument has been fully considered but has been found unpersuasive.

6. Applicants also argue that:

With respect to Claim 4, Applicants respectfully assert that Rosenberg or Coker or their combination fails to teach or suggest the limitation that the microcontroller and the virtual microcontroller are pointing to the same assembly instruction line 0 after executing the boot code. Applicants respectfully assert that the combined references are completely silent as to the limitation.

The Examiner respectfully traverses this rejection as follows.

Applicants' claim language must not be interpreted literally because doing so renders the invention inoperable. It is widely known to persons of ordinary skill in the art that a processor does not directly execute assembly code but rather executes machine code. This distinction is critical, especially when describing a processor at the level of timing instructions (claim 1), halting processors (claim 1), copying registers (claim 2), branching instructions (claim 4), setting break points (claim 5), et cetera.

Therefore, the claim language "the microcontroller branches to an assembly instruction line 0" cannot be interpreted literally while simultaneously having an invention that complies with 35 U.S.C. § 112.

A person having ordinary skill in the art would interpret this limitation as “the microcontroller branches to the beginning of an unspecified program” because “assembly line 0” refers to the assembly instruction that corresponds to the beginning of the machine instructions which will be executed for that program. (Applicants’ specification provides no explanation whatsoever of any significance of “assembly instruction line 0” as opposed to “assembly instruction line 0xFF” or any other arbitrarily chosen assembly instruction line.)

Rosenberg teaches that after executing code, a processor branches to an instruction line [*“A breakpoint is a special code placed in the executing code stream that, when executed, causes a special trap to occur that the processor and the operating system report to the debugger.”* (page 5, second full paragraph); More generally at pages 56-57, “Generic OS-Debugger Interaction Model”, etc.].

Therefore, after executing the boot code and halting (taught in the rejection of claim 1), Rosenberg teaches “branching” (a special trap) to an assembly line instruction 0 (the beginning of the “processor and the operating system report”).

Applicants’ argument has been fully considered but has been found unpersuasive.

7. Applicants also argue that:

Rosenberg or Coker or their combination fails to teach or suggest the limitation that a breaker is set at an assembly instruction line 0 prior to the executing the boot code, and prior to the executing the timing code.

This argument has been addressed above and found unpersuasive.

8. Applicants also argue that claim 7 combines the limitations of claims 4 and 5, while claim 8 combines the limitations of claims 2, 4, and 5, and each should be allowable for the reasons set forth above. These arguments have been addressed above and found unpersuasive.

9. Applicants also argue that:

With respect to currently amended claim 20, Applicants respectfully assert that Rosenberg or Coker or their combination fails to teach or suggest the limitation of removing the breaker at assembly instruction line 0 after the copying the register contents and the copying the memory contents.

The Examiner respectfully traverses this argument as follows.

Applicants' argument is unpersuasive in light of the teachings of "How Debuggers Work" by Rosenberg. Pages 98-99 of Rosenberg provide a basic introduction to what is known to persons of ordinary skill in the art regarding breakpoints.

Applicants' argument has been fully considered but has been found unpersuasive.

10. Applicants also argue that:

With respect to previously presented claim 22, Applicants respectfully assert that Rosenberg, Coker, Matyas or their combination fails to teach or suggest the limitation of "the boot code containing algorithms."

The Examiner respectfully traverses this argument as follows.

Without algorithms, it would be boot data, not boot code.

Applicants' argument has been fully considered but has been found unpersuasive.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. § 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. § 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. § 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. § 103(c) and potential 35 U.S.C. § 102(e), (f) or (g) prior art under 35 U.S.C. § 103(a).

11. Claims 1-2, 4-10, 12, 14-20, and 23 are rejected under 35 U.S.C. § 103(a) as being unpatentable over US Patent No. 5,371,878 to Coker in view of “How Debuggers Work” by Jonathan B. Rosenberg (Rosenberg), further in view of US Patent No. 5,968,135 to Teramoto et al.

Regarding claim 1, Coker teaches a method comprising:

In a microcontroller, executing a set of boot code to carry out initialization (“Embedded Computer System or ECS”, see column 1, lines 20-23) [*“[T]his invention uses hardware and software which can ‘shadow’ the execution of a target-ECS in real time operation”* (column 2, lines 34-40); *“A shadow system of this invention executes the same software as the target-ECS from system start-up or reset.”* (column 2, lines 56-58); Applicants’ remarks state that: “Booting is a process that starts a system (e.g., operating system) and substantially initialize the system when a user turns on a computing system or a system with a processor.” (Applicants’ remarks, 6 February 2007, page 15)];

In a virtual microcontroller (“shadow system”), executing a set of timing code to enable the lock-step synchronization [*“A shadow system of this invention executes the same software as the target-ECS from system start-up or reset.”* (column 2, lines 56-58); *“The shadow system and the target-ECS function exactly the same except that the shadow system receives data slightly delayed because of a data buffer between the target-ECS and the shadow system.”* (column 3, lines 13-16); *“[I]n terms of relative time, the execution state of the shadow system 28 at the time when any given instruction is executed will directly correspond to the execution state of the target-ECS when the same instruction was executed..”* (column 8, lines 44-49)]

and wherein the set of timing code is different from said set of boot code, [*“Thus, rather than receiving input data from an external source and reading the data into complex I/O registers, the shadow system uses the data value and relative time of input events from the target-ECS and writes the value directly to its RAM using its internally generated location.”* (column 2, line 63 – column 3, line 1)];

and wherein the set of boot code is stored within the microcontroller and the set of boot code is inaccessible to the virtual microcontroller [Interface means 19 connecting the target-ECS and shadow system is used by Coker to transmit I/O data and does not appear to contain any suggestion that instruction code or boot code is transmitted via interface means. See (column 4, lines 9-18)].

Although Coker teaches “a computer system and method for debugging, verifying and developing an embedded computer system” (column 1, lines 9-11), Coker does not expressly teach old and well-known tools and techniques of debugging. As a result, Coker teaches a method of debugging, but does not expressly teach simultaneously halting both the microcontroller and the virtual microcontroller.

Rosenberg expressly teaches halting multiple processors operating in lockstep [*A significant issue for debuggers on multiprocessor systems is how or whether other processors stop when a fault occurs in one. On SIMD architectures, by definition, all processors operate in lock step and so this is guaranteed.*” (page 45, first full paragraph)].

Rosenberg and Coker are analogous art because both are drawn to debugging.

It would have been obvious to a person of ordinary skill in the art at the time of Applicants’ invention to combine the teachings of Rosenberg and Coker because Rosenberg provides teachings specifically directed to debugging tools, which are suggested by Coker, but “are very difficult tools to build robustly because they depend heavily on relatively weak portions of operating systems and because they tend to stress the underlying operating system’s capabilities.” (Rosenberg, page 1, first paragraph). That is, Rosenberg provides the teachings that provide helpful guidance in making and using the “critical tools for the development of

software" (Rosenberg, page 1, first paragraph) that are suggested by Coker (title, abstract). The combination could be achieved as expressly taught by Rosenberg, by halting the target-ECS and the shadow system "simultaneously" because Rosenberg explicitly teaches this behavior in architectures, like the one claimed, that operate in lock step. (*Id.*)

Coker in view of Rosenberg does not expressly teach that the timing code is a dummy code timed to take the same number of clock cycles as the microcontroller uses to execute the set of boot code. Applicants have defined "dummy code" as exemplified in the remarks submitted on 31 October 2007.

Teramoto teaches timing code that is a dummy code timed to take the same number of clock cycles as a second processor executing code [*The present invention relates to an instruction execution control method and information processing apparatus for monitoring information about the completion of synchronization between processors, and selectively causing a specific instruction of the subsequent group of instructions to wait until the completion of synchronization is indicated when the processors are operated in synchronism with each other for synchronous execution of their respective processes in a computer system including a plurality of processors.*" (column 1, lines 7-16); "*When the instruction, which was read out, is a Wait instruction, the instruction analyzer 102 issues a Wait instruction to the Wait instruction controller 100 through the interface signal 105. The Wait instruction controller 100 executes the Wait instruction which controls the instruction execution sequence according to the present invention.*" (column 5, lines 50-60)].

Teramoto and Rosenberg in view of Coker are analogous art because both are drawn to operating processor synchronously.

It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the teachings of Teramoto with Rosenberg in view of Coker as expressly motivated by Teramoto to maintain high execution speed [*"An object of the present invention is to provide an instruction execution control method and an information processing apparatus for enabling synchronized operations to be performed at high speed among a plurality of processors sharing a main storage."* (column 2, lines 40-44)].

Therefore it would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the teachings of Teramoto with Rosenberg in view of Coker to achieve the invention specified in claim 1.

Regarding claim 2, Coker teaches copying register contents from the microcontroller to corresponding registers in the virtual microcontroller after completion of the simultaneous halting [*"An execution state, including the contents of RAM and I/O registers in the target-ECS and RAM and I/O state memory in the shadow system, can further be defined as including an input state vector, output state vector, and internal state vector. Input and output state vectors are defined as the contents of input and output registers respectively in the target-ECS and as the contents of the I/O memory state in the shadow system."* (column 8, lines 50-57); copied from the target-ECS to the shadow system, "*the shadow system receives its input data from the input registers of the target-ECS and stores the input data in its RAM.*" (column 2, lines 61-63)].

Also, Rosenberg teaches that "*debuggers provide disassembly views and direct access to hardware registers.*" (page 6, first full paragraph).

Regarding claim 4, Rosenberg teaches that after executing code, a processor branches to an instruction line [*A breakpoint is a special code placed in the executing code stream that, when executed, causes a special trap to occur that the processor and the operating system report to the debugger.*” (page 5, second full paragraph); More generally at pages 56-57, “Generic OS-Debugger Interaction Model”, etc.].

Regarding claim 5, Rosenberg teaches that prior to executing code, a break is set at an assembly instruction line [*A breakpoint is a special code placed in the executing code stream that, when executed, causes a special trap to occur that the processor and the operating system report to the debugger.*” (page 5, second full paragraph)].

Regarding claim 6, Coker teaches that the boot code comprises protected initialization code that is not accessible to the In-Circuit Emulation system [Coker teaches no means by which the in-circuit emulation system may access the code on the target-ECS. Here Applicants’ have presented a negative limitation, which has been interpreted and treated according to MPEP 2173.05(i).].

Claim 7 recites a combination of limitations recited by claims 4 and 5, which are taught by Coker in view of Rosenberg, further in view of Teramoto as shown above.

Claim 8 recites a combination of limitations recited by claims 2, 4, and 5, which are taught by Coker in view of Rosenberg, further in view of Teramoto as shown above.

Regarding claim 9, Rosenberg teaches removing a break at an assembly line [*"Inactive breakpoints are place holders that the user can turn active but currently will not cause execution to stop if reached."* (page 27)].

Claim 10 recites a combination of limitations recited by claims 1, 2, and 9. As Coker in view of Rosenberg, further in view of Teramoto teaches these claims, claim 10 is rejected for similar rationale.

Claims 12-20 present a broader recitation of claims 1-9. These claims are rejected for rationale similar to that given above for claims 1-9 as being obvious over Coker in view of Rosenberg, further in view of Teramoto.

Regarding claim 23, Coker teaches that at least one portion of the boot code is stored internally in said microcontroller [*"In other words, an ECS normally executes software..."* (column 1, lines 29-39); As noted above, Coker teaches no means by which the in-circuit emulation system may access the code on the target-ECS.]

12. Claim 21 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Coker in view of Rosenberg, further in view of Teramoto as applied to claim 12 above, and further in view of "Emulation of the Sparcle Microprocessor with the MIT Virtual Wires Emulation System" by

Matthew Dahl, Jonathan Babb, Russel Tessier, Silvina Hanono, David Hoki, and Anant Agarwal (Dahl) and further in view of “A Reconfigurable Logic Machine for Fast Event-Driven Simulation” by Jerry Bauer, Michael Bershteyn, Ian Kaplan, and Paul Vyedin (Bauer).

Coker in view of Rosenberg, further in view of Teramoto does not expressly teach that the shadow system is implemented on a field programmable gate array (FPGA).

Dahl teaches that it is known in the art to emulate a Sparc microprocessor using an FPGA (abstract).

Bauer teaches that hardware emulation can increase simulation speed by up to 10,000 times (introduction, paragraphs 1-2).

Therefore it would have been obvious to a person of ordinary skill in the art at the time of Applicants’ invention to combine these teachings and arrive at the decision to implement the shadow system of Coker on an FPGA to realize an enormous increase in simulation speed. Knowledge that this was possible is provided by Dahl, and motivation to combine the references, to increase simulation speed, is provided by Bauer.

Therefore it would have been obvious to a person of ordinary skill in the art at the time of Applicants’ invention to combine the teachings of Coker in view of Rosenberg, further in view of Teramoto with Dahl and Bauer to arrive at the invention specified in claim 21.

13. Claim 22 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Coker in view of Rosenberg, further in view of Teramoto as applied to claim 1 above, and further in view of US Patent No. 4,757,534 to Matyas et al.

Regarding claim 22, Coker in view of Rosenberg, further in view of Teramoto does not disclose that the boot code comprises serial numbers, passwords, and algorithms.

Matyas teaches code that comprises serial numbers, passwords, and algorithms [*In carrying out the computer/smart card protocol, the T output is sent to the smart card together with the parameters P1 (password) and P2 (program number | diskette serial number, where | denotes concatenation) and a third parameter P3. Where the crypto facility uses the DES algorithm, as shown in FIGS. 8 and 9, to encrypt the file key, P3 is the computer number.*” (column 13, lines 1-21)].

Matyas and Coker in view of Rosenberg, further in view of Teramoto are analogous art because both are directed to computer software.

It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the teachings of Matyas with Coker in view of Rosenberg, further in view of Teramoto to include the passwords, serial numbers, and algorithms in the boot code in order to discourage “the copying and sharing of purchased software program” (here, the boot code) as expressly taught by Matyas (abstract). The combination could be achieved by implementing the smart card cryptographic method taught by Matyas with the target-ECS system taught by Coker.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the teachings of Matyas with Coker in view of Rosenberg, further in view of Teramoto to arrive at the invention specified in claim 22.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Proctor whose telephone number is (571) 272-3713. The examiner can normally be reached on 8:30 am-4:30 pm M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Paul Rodriguez can be reached at (571) 272-3753. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Jason Proctor
Examiner
Art Unit 2123

jsp


ZOILA CABRERA
PRIMARY EXAMINER
TECHNOLOGY CENTER 2100
1/28/08